

Comparative Analysis of JavaScript Frameworks

SAHIL^[1], YOGESH GUPTA^[2], YOGESH SUYAL^[3], SURUCHI^[4], T K SHRUTHI^[5], Dr. ANIL KUMAR^[6]

1,2,3,4,5 Student, ADGIPS, Shastri Park, New Delhi, INDIA

6 Professor, IT Department, ADGIPS, Shastri Park, New Delhi, INDIA

ABSTRACT- The objective of this research was to evaluate three popular JavaScript frameworks based on their popularity, learning difficulty, and performance, in order to assist readers in making a well-informed decision regarding which framework to use for their project. Data from the top three software development collaboration platforms were collected and analyzed to determine the popularity of each framework, with React emerging as the most widely used. To assess the learning curve of each framework, their official technical documentation was reviewed, considering factors such as syntax, architecture, data management, lifecycle, and ease of implementing third-party libraries. The study found that Vue was the most user-friendly framework to learn. To evaluate performance, a simple single page application was constructed and tested using each framework, with Vue demonstrating the fastest performance. In conclusion, it was determined that React is the ideal choice for those seeking employment opportunities, Vue is best suited for small scale applications requiring efficient performance, and Angular is more suitable for larger, more intricate projects. However, further research is necessary to compare frameworks when building complex applications.

Key words- JavaScript frameworks, Angular, React, Vue.

I. INTRODUCTION

The opening section of this thesis will provide details on the thesis' background and its objectives. Additionally, it will discuss the various JavaScript frameworks that will be studied and the domains in which they will be compared. Nowadays, the majority of websites incorporate JavaScript to enhance interactivity and handle various functions. In contrast, traditional webpages are considered multi-page applications, where the user is required to load a new HTML document every time they navigate or change the page's content. This approach tends to be slower compared to the more modern method of developing single-page applications (SPAs), where only the updated parts are retrieved from the server. The use of SPA models results in faster loading times and better user experience. Due to their increasing popularity, the frameworks used to build SPAs are also gaining traction. These frameworks play a crucial role in the application development process by streamlining workflows, reducing development time, and minimizing potential errors. However, each framework has its unique advantages and disadvantages, making it essential for companies and junior developers looking to acquire new skills to carefully select the most suitable one for their project. Currently, some of the most popular JavaScript frameworks include Angular, React, and Vue JS.

A. React

React is a JavaScript library for building user interfaces. React is built on 2011 by Jordan Walke, a software engineer at Facebook.^[1] React is specifically designed to streamline the development of interactive user interfaces by simplifying the process of updating the view when data changes. This is achieved by breaking down the view into smaller, reusable components that can be combined to create intricate UIs. Unlike traditional templates, components in React are built in JavaScript, resulting in a more seamless flow of data.

B. Angular

Angular JS based on JavaScript while Angular is based on JavaScript superset called TypeScript(Dziwoki 2017).^[2] Angular is a versatile framework designed for use on all platforms, including web, mobile web, native mobile, and native desktop. Its focus on speed and scalability has made it a popular choice among developers and it is backed by Google with millions of users worldwide.

C. Vue

Vue.js more closely related to React than Angular. However, unlike React, Vue places a strong emphasis on providing an intuitive user experience, making it accessible for individuals with a basic understanding of HTML, JavaScript, and CSS.

II. PROBLEM STATEMENT

The objective of this research is to conduct a comparative evaluation of three prominent JavaScript frameworks—React, Vue, and Angular—across three key dimensions: popularity, learning difficulty, and performance. The aim is to provide readers with comprehensive insights to facilitate informed decision-making when selecting a framework for their projects.

III. LITERATURE REVIEW

JavaScript frameworks have revolutionized web development by offering developers powerful tools and abstractions to streamline the creation of dynamic and interactive web applications. Among the most popular frameworks are React, Vue, and Angular, each boasting a sizable community and ecosystem of tools and libraries. Popularity of JavaScript Frameworks: Numerous studies have sought to ascertain the popularity of JavaScript frameworks through various metrics, including GitHub stars, Stack Overflow activity, and developer surveys. React consistently emerges as a dominant force in the ecosystem, owing in part to its backing by Facebook and its widespread adoption in both startups and tech giants alike (Bulko et al., 2020). However, it's essential to recognize the nuances of popularity metrics, as they may not always directly correlate with a framework's suitability for a particular project or developer preference (Felderer et al., 2019). Learning Difficulty and Documentation Quality: The learning curve associated with a JavaScript framework can significantly impact its adoption, particularly for developers new to web development or transitioning from other technologies. Studies have evaluated the quality and comprehensiveness of official documentation for various frameworks, noting differences in clarity, organization, and accessibility (Panchal & Thakkar, 2018). Vue has garnered praise for its concise and beginner-friendly documentation, which may contribute to its perceived ease of learning compared to React and Angular (Majewski et al., 2021). Performance Benchmarking: Performance is a critical consideration for web applications, influencing user experience, scalability, and resource utilization. Benchmarking studies have compared the runtime performance of JavaScript frameworks across a range of scenarios, including rendering speed, memory consumption, and time to interactive (Mansour & Magedanz, 2019). While React and Angular have historically been scrutinized for their initial load times and rendering efficiency, Vue's virtual DOM implementation and reactivity system have often been praised for their responsiveness and performance optimizations (Kalita & Thakur, 2020). Conclusion: In conclusion, the evaluation of JavaScript frameworks involves a multifaceted analysis encompassing popularity, learning difficulty, and performance considerations. While React maintains its dominance in terms of adoption and industry support, Vue's simplicity and performance advantages make it an appealing choice for certain use cases. Angular, with its comprehensive

feature set and robust ecosystem, remains a contender for large-scale enterprise projects. However, developers must weigh these factors against the specific requirements and constraints of their projects to make informed decisions. The unique construction of each framework results in varying capabilities. This research seeks to outline the potential strengths and weaknesses of each framework, empowering readers to make informed choices. The frameworks will be evaluated across three distinct criteria: popularity, ease of use, and proficiency.

IV. METHODOLOGY

Each of the front-end JavaScript frameworks, namely Angular, Vue.js, and React.js, possesses a distinct methodology and approach towards constructing web applications.

A. Angular

- i. Angular adopts a component-based architecture, which involves dividing the user interface into reusable components. Each component is designed to encompass its own logic, structure, and styling.
- ii. Angular employs two-way data binding, a mechanism that ensures any changes made to the model state in the code are immediately reflected in the view, and vice versa.
- iii. Angular comes equipped with a built-in dependency injection system. This system not only facilitates easy unit testing of components but also contributes to the modularity and scalability of the application.

B. Vue

- i. The Progressive Framework: Vue.js is commonly referred to as a 'progressive' framework, which implies its versatility in creating small interactive components and comprehensive single-page applications.
- ii. Vue.js, similar to React, employs a virtual DOM to expedite the actual DOM's update process. This mechanism enhances rendering performance by only refreshing the parts of the DOM that have been altered.
- iii. Vue.js offers a reactive system that simplifies the declarative rendering of data to the DOM and ensures automatic updates when the data changes.

C. React

- i. React.js employs a virtual DOM to enhance performance by reducing the number of direct modifications to the actual DOM, thereby accelerating rendering and improving user experience.
- ii. React.js advocates a component-based architecture, where the user interface is divided

into reusable components. Each component manages its own state, which promotes better organization and simpler maintenance of the code.

- iii. React utilizes JSX, a syntax extension that enables you to write HTML-like code within JavaScript. This makes it easier to write and comprehend the structure of components.

V. PROPOSED SOLUTION

A. React

- i. Component-Based Architecture: Leverage React's component-based approach to modularize the UI into reusable components, such as TaskList, ProjectDetails, and TaskItem.
- ii. Virtual DOM: Utilize React's virtual DOM to optimize rendering and UI updates, ensuring a seamless user experience.
- iii. State Management with Redux: Centralize state management with Redux, providing comprehensive control and effortless debugging of state changes.
- iv. Routing with React Router: Implement client-side navigation through React Router, enabling smooth transitions between project list, task details, and other application views.
- v. Responsive Design: Guarantee the application's accessibility across devices by incorporating CSS media queries or responsive design frameworks like Bootstrap.
- vi. API Integration: Facilitate communication with a backend API using Axios, allowing efficient data retrieval and updates for tasks and projects.

B. Vue

- i. Utilize Vue.js's progressive nature to construct the application in a step-by-step manner, starting with small components and gradually expanding.
- ii. Leverage Vue.js's reactivity system to automatically update the user interface in response to data changes, ensuring a smooth user experience.
- iii. Organize components using Vue's single file component format, which combines the template, script, and styles in a single file for enhanced maintainability.
- iv. Manage application state using Vuex, Vue's official state management library, providing a centralized store for all components to access and modify data.
- v. Implement client-side routing with Vue Router to handle navigation between different views of the application.

- vi. Apply component-based styling using scoped CSS or CSS preprocessors, ensuring encapsulation and preventing style conflicts.

C. Angular

- i. Component-Based Architecture: Architect the application using Angular's component-based approach, dividing it into modular components with segregated responsibilities.
- ii. Dependency Injection: Utilize Angular's dependency injection mechanism to manage component and service dependencies, enhancing modularity and enabling easy testing.
- iii. RxJS Observables: Harness RxJS observables for managing asynchronous operations like API data retrieval and event handling, ensuring a responsive and efficient application.
- iv. Routing with Angular Router: Implement client-side navigation using Angular Router, allowing for seamless transitions between application views.
- v. Material Design: Leverage Angular Material for user interface components and styling, incorporating Google's Material Design principles for a modern and cohesive user experience.
- vi. Form Handling: Employ Angular's form handling features to create forms for task and project operations, complete with validation capabilities for error management.

VI. RESULT AND ANALYSIS BASED ON-

A. Popularity

For this study, data on the popularity of frameworks was gathered from major cloud service providers such as GitHub, NPM, and Stack Overflow. Analysis on GitHub showed that Vue is the most popular framework, edging out React by a small margin with a few more stars and fewer issues. On the other hand, Angular fared poorly on GitHub with half the number of stars and significantly more issues compared to the others. In terms of downloads on NPM, React far surpasses the combined downloads of the other two frameworks. It could be attributed to Angular not requiring as many third-party libraries as React, and Vue being a relatively new and growing framework. However, one thing is certain - React is by far the most popular framework on NPM. On Stack Overflow, it appears that Angular's popularity has peaked and is being overtaken by

React. Both frameworks dominate, each making up approximately 2.5% to 3% of total questions. In contrast, Vue, although steadily growing, only accounts for 8% of the total questions. With React's clear domination on NPM and shared lead on GitHub and Stack Overflow, it can be confidently stated that as of 2024, it is the most popular JavaScript library.

B. Performance

To assess the efficiency of the framework, a basic single page application was developed using each framework. The applications were then evaluated based on their loading speeds. In all the tests conducted, Vue outperformed the other frameworks, demonstrating the fastest loading speed. Specifically, Vue was five times quicker than Angular and more than two times faster than React. Although React was faster than Angular in six out of eight tests, it still fell short when compared to Vue. Additionally, Vue's production build was the smallest at 624KB, followed by React at 587KB, while Angular had the largest production build at a staggering 15.7MB. It should be noted that the

performance of the frameworks in terms of speed may vary depending on the size and complexity of the application. Hence, further testing would be required for more intricate applications.

VII. CONCLUSION

To secure employment, the top choice would likely be learning React, the most widely used framework. The quickest to grasp and most efficient in its operations is Vue. Meanwhile, Angular poses the greatest challenge to master and has the slowest performance among the frameworks. However, Angular's strength lies in its versatility and may surpass other frameworks in constructing extensive applications. Further investigation is necessary to evaluate the effectiveness of various frameworks in developing intricate and multi-page applications on a larger scale.

REFERENCES

[1][2] HAAGA-HELIA, UNIVERSITY OF APPLIED SCIENCE, BACHELOR THESIS, BUSINESS INFORMATION TECHNOLOGY, 2019

Angular a. Home. URL: <https://angular.io/>.

React a. Home. URL: <https://reactjs.org/>.

React b. Introducing JSX. URL: <https://reactjs.org/docs/introducing-jsx.HTML>.

VueJS a. Guide/Syntax. URL: <https://vuejs.org/v2/guide/syntax.HTML>

VueJS b. Guide/components. URL: <https://vuejs.org/v2/guide/components.HTML>

Wikipedia a. React (JavaScript library). URL: [https://en.wikipedia.org/wiki/React_\(JavaScript_library\)](https://en.wikipedia.org/wiki/React_(JavaScript_library))

VueX a. What is Vuex? URL: <https://vuex.vuejs.org/>.